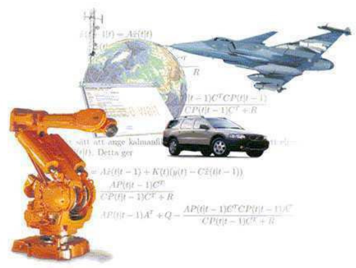
Sysid Course VT1 2016
Nonlinear Models



Lennart Ljung

Automatic Control, ISY, Linköpings Universitet

Nonlinear models - Outline

- General aspects
- Black-box models: Neural network models and the like
- Grey-box models: Physical, Block-oriented, Local models

General Aspects

→ 148

Let Z^t denote all available (input-output) data up to time t. A mathematical model for the system is a function from these data to the space where the output at time t, $y(t)$ lives, in general

$$\hat{y}(t|t-1) = g(Z^{t-1}, t)$$

The function can be thought of as a predictor of the next output.
A parametric model is a parameterized family of such models:

$$g(Z^{t-1}, \theta)$$

The difficulty is the enormous richness in possibilities of parameterizations.There are two main cases

- Black-box models: General models of great flexibility
- Grey-box models: Models that incorporate some knowledge of the character of the actual system.

Nonlinear models - Outline

- General aspects
- Black-box models: Neural network models and the like
- Grey-box models: Physical, Block-oriented, Local models

Black-box Models: General Comments

→ 148

The general mapping $g(Z^{t-1}, \theta)$ is normally too flexible. Let us split it into one mapping from Z^{t-1} to a regression vector $\varphi(t)$ of fixed dimension d and a mapping g from R^d to R (assuming the output to be scalar):

$$g(Z^{t-1}, \theta) = g(\varphi(t), \theta)$$
$$\varphi(t) = \varphi(Z^{t-1}) \quad (\text{ or } \varphi(t, \theta) = \varphi(Z^{t-1}, \theta))$$

Leaves two problems

1. Choose the mapping $g(\varphi, \theta)$
2. Choose the regression vector $\varphi(t)$

NL Black Box: Choice of Regressors φ

→ 149

Four players:

- Inputs $u(t-k)$
- Outputs $y(t-k)$
- Simulated model outputs $\hat{y}_s(t-k, \theta)$
- Predicted model outputs $\hat{y}_p(t-k|\theta)$

Regressors for dynamical systems are typically chosen among the first ones:

- NLFIR-models use past inputs
- NLARX-models use past inputs and outputs
- NLOE-models use past inputs and past simulated outputs
- NLARMAX-models use inputs, outputs and predicted outputs
- NLBJ-models use all four regressor types
- NLARX is the dominating model

NL Black Box: Choice of g

→ 149

First, consider φ to be scalar. Basic basis function expansion: $g(\varphi, \theta) = \sum_{k=1}^{N} \alpha_k \kappa_k(\varphi)$. Typical case: $\kappa_k(\varphi) = \kappa(\beta_k(\varphi - \gamma_k))$

$$g(\varphi, \theta) = \sum_{k=1}^{N} \alpha_k \kappa(\beta_k(\varphi - \gamma_k))$$

- $\kappa(x) = \cos(x)$: Fourier transform
- $\kappa(x) = U(x)$: Unit pulse, gives piecewise constant functions g.
  - Soft version: $\kappa(x) = e^{-x^2/2}$
- $\kappa(x) = H(x)$: Step at $x = 0$, gives also piecewise constant functions
  - Soft version: $\kappa(x) = \frac{1}{1+e^{-x}}$
- α coordinates, β scale or dilation, γ location

Several Regressors

→ 150

Consider now φ to be a d-dimensional vector, but let still $\kappa(x)$ be a function of one variable. How to interpret $\kappa(\beta(\varphi - \gamma))$?

Radial $\beta(\varphi - \gamma) = \|\varphi - \gamma||_\beta = (\varphi - \gamma)^T \beta(\varphi - \gamma)$
γ a d-dimensional vector, β a $d|d$-matrix (positive definite) or scaled version of the identity matrix with β a scalar.
Describes an ellipsoid in R^d.

Ridge $\beta(\varphi - \gamma) = \beta^T \varphi - \gamma$
β a d-dimensional vector, γ a scalar.
Describes a hyperplane in R^d

Tensor κ is a product of factors corresponding to the components of the vector:
$\kappa(\beta(\varphi - \gamma)) = \prod_{k=1}^{d} \kappa(\beta_k(\varphi_k - \gamma_k))$
γ and β are d-dimensional vectors and subscript denotes component.

Examples of Named Structures

→ 155

- ANN: artificial Neural Networks
  - One hidden layer sigmoidal: $\kappa(x) = \frac{1}{1+e^{-x}}$, ridge extension
  - Radial Basis Networks: $\kappa(x) = e^{-x^2/2}$, radial extension
- Wavelets: κ is the "mother wavelet" and $\beta_j = 2^j$, $\gamma_k = 2^{-j}k$ (double indexing) as fixed choices
- (Neuro)-Fuzzy models: κ are the membership functions, tensor expansion

Simulation and Prediction

Suppose $\varphi(t) = [y(t-1), u(t-1)]^T$
The (one-step ahead) predicted output at time for a given model θ is then

$$\hat{y}_p(t|\theta) = g([y(t-1), u(t-1)]^T, \theta)$$

It uses the previous measurement $y(t-1)$.

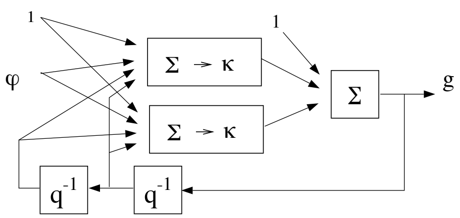
A tougher test is to check how the model would behave in simulation, i.e. when only the input sequence u is used. The simulated output is obtained as above, by replacing the measured output by the simulated output from the previous step:

$$\hat{y}_s(t,\theta) = g([\hat{y}_s(t-1,\theta), u(t-1)]^T, \theta)$$

Notice a possible stability problem!

Recurrent Networks

For NLOE, NLARMAX and NLBJ, previous outputs from the model have to be fed back into the model computations on-line:

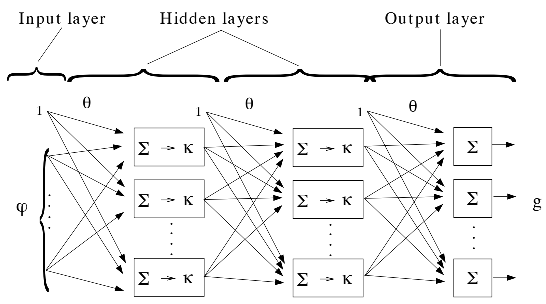


These are called recurrent networks and require considerable more computational work to fit to data.

Network Aspects – Several Layers

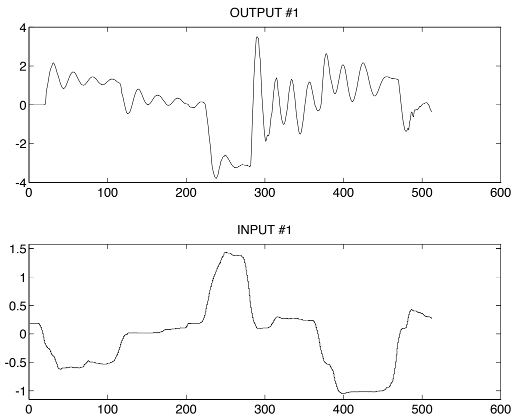
→ 153

The model structures are really basis function expansions. However, since the basis functions are variants of the same function κ, a graphical description looks like a network. One can also let the regressors be outputs from a previous layer of the network: ("deep learning")



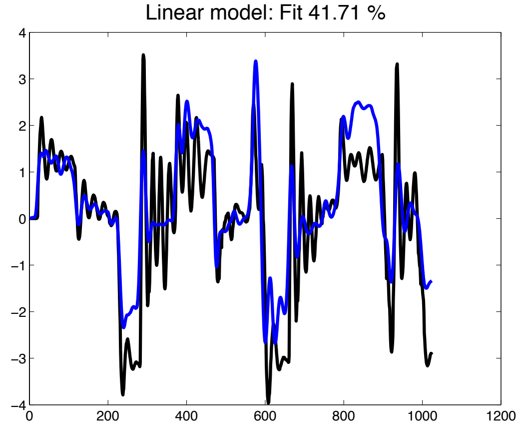
Example: Hydraulic Crane Data

These are data from a forest harvest machine:

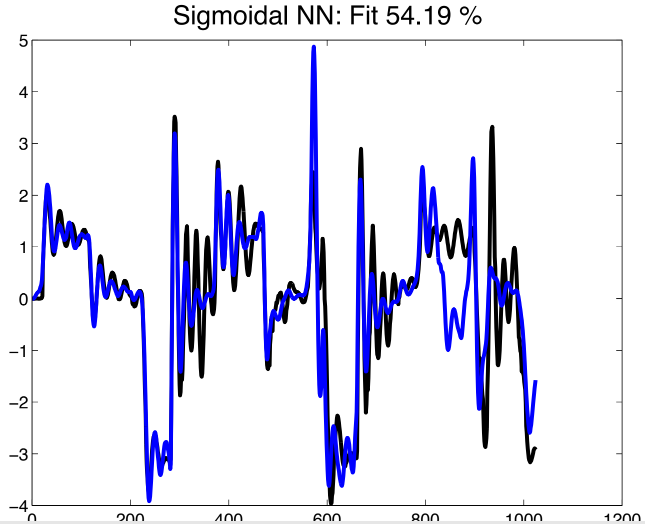


Linear Model

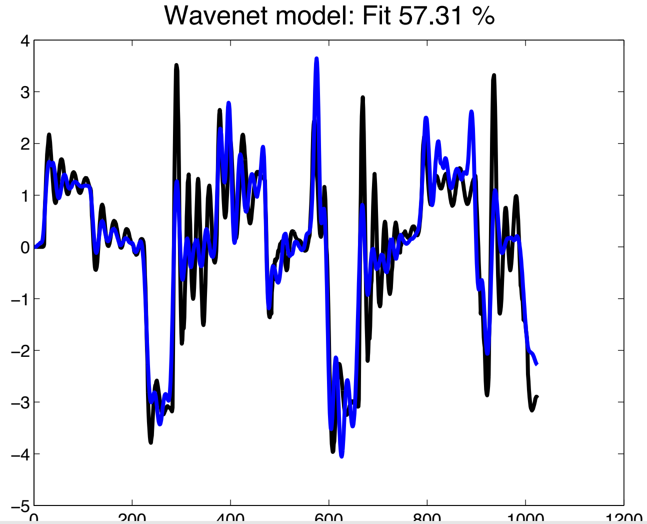
Black: Measured Output
Blue: Model Simulated Output



Sigmoidal ANN model



Wavenet Model (Radial BF ANN Model)



Nonlinear models - Outline

- General aspects
- Black-box models: Neural network models and the like
- Grey-box models: Physical, Block-oriented, Local models
  - Physical Modelling
  - Semiphysical Modelling
  - Block-oriented models
  - Local Linear Models

Physical Modeling

→ 146

Perform physical modeling (e.g. in MODELICA) and denote unknown physical parameters by θ. Collect the model equations as

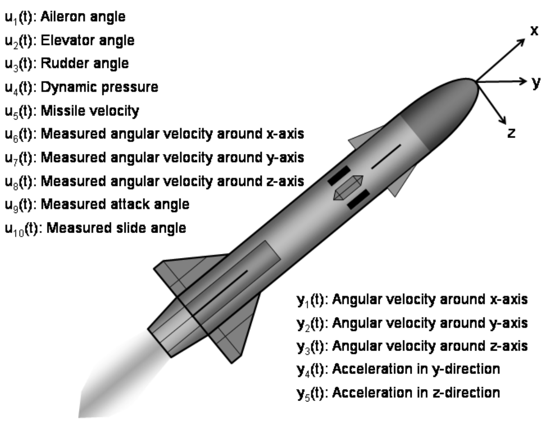
$$\dot{x}(t) = f(x(t), u(t), \theta)$$
$$y(t) = h(x(t), u(t), \theta)$$

(or in DAE, Differential Algebraic Equations, form.) For each parameter θ this defines a simulated (predicted) output $\hat{y}(t|\theta)$ which is the parameterized function

$$\hat{y}(t|\theta) = g(Z^{t-1}, \theta)$$

in somewhat implicit form. To be a correct predictor this really assumes white measurement noise. Some more sophistical noise modeling is possible, usually involving *ad hoc* non-linear observers. The approach is conceptually simple, but could be very demanding in practice.

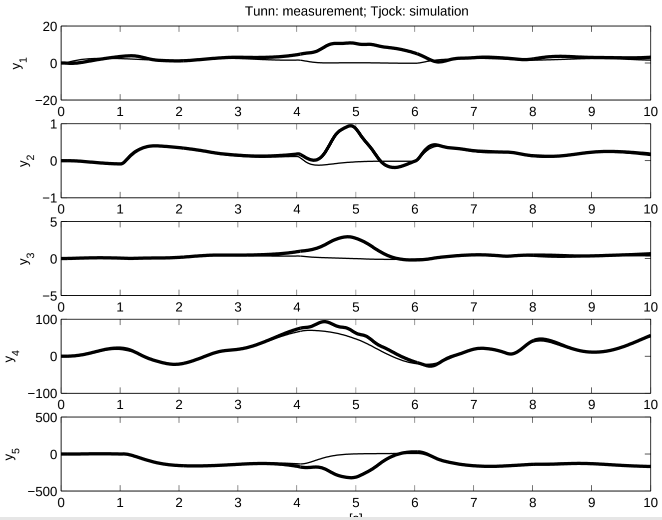
Example: Missile



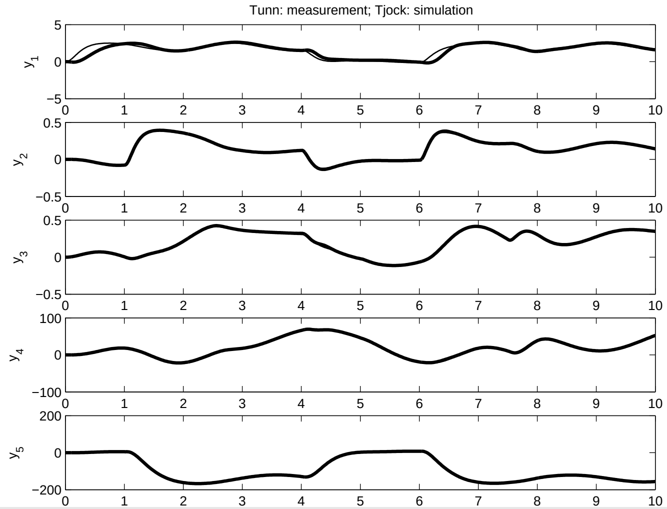
The Equations

function [dx, y] = missile(t, x, p, u);
MISSILE A non-linear missile system.
Output equation. y = [x(1); ... x(2); ... x(3); ...
-p(18)*u(4)*(p(1)*x(5)+p(2)*u(3))/p(22); ...
-p(18)*u(4)*(p(3)*x(4)+p(4)*u(2))/p(22) ...];
State equations. dx =
[1/p(19)*(p(17)*p(18)*(p(5)*x(5)+0.5*p(6)*p(17)*x(1)/u(5)+ ... %
Angular velocity around x-axis.
p(7)*u(1))*u(4)-(p(21)-p(20))*x(2)*x(3))+ ... p(23)*(u(6)-x(1)); ...
1/p(20)*(p(17)*p(18)*(p(8)*x(4)+0.5*p(9)*p(17)*x(2)/u(5)+ ... ...
p(1)-p(25) unknown parameters u, y : measured inputs and outputs

Initial Fit between Model and Data



Adjusted Fit between Model and Data



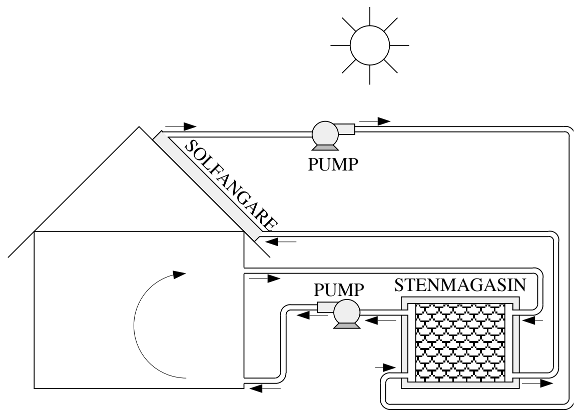
Semi-physical Models

→ 145

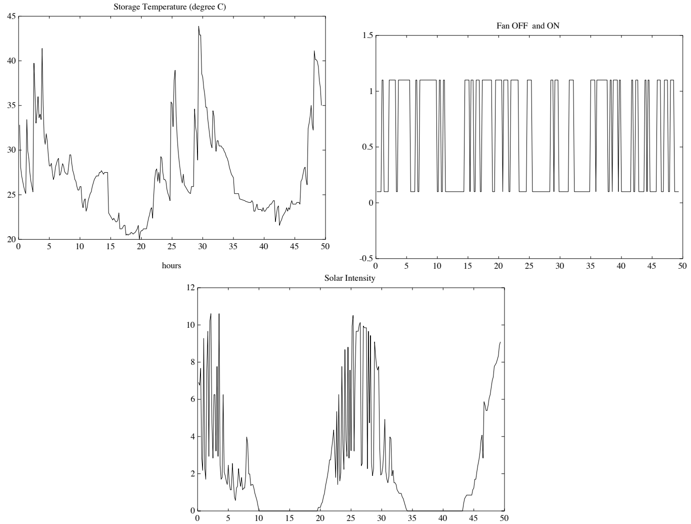
Apply non-linear transformations to the measured data, so that the transformed data stand a better chance to describe the system in a linear relationship.
"Rules: Only high-school physics and max 10 minutes"
Simple examples: . . ..
Another example:...

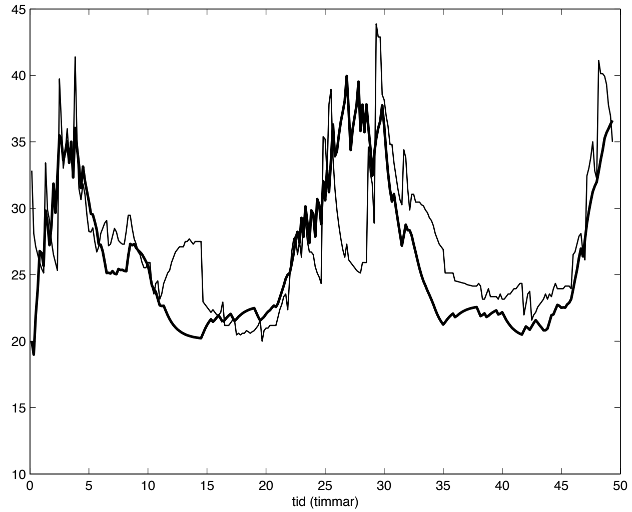
A Solar Heated House

→ 145



$y(t)$: temperature in storage; $I(t)$: Solar intensity; $u(t)$: Pump speed





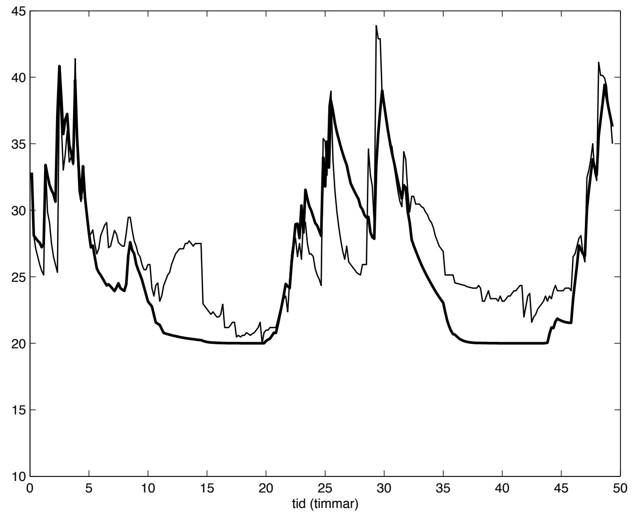
Suppose we had measured the temperature $x(t)$ in the solar panel:

$$x(t+1) - x(t) = d_2 I(t) - d_3 x(t) - d_0 x(t) \cdot u(t)$$
$$y(t+1) - y(t) = d_0 x(t) \cdot u(t) - d_1 y(t)$$

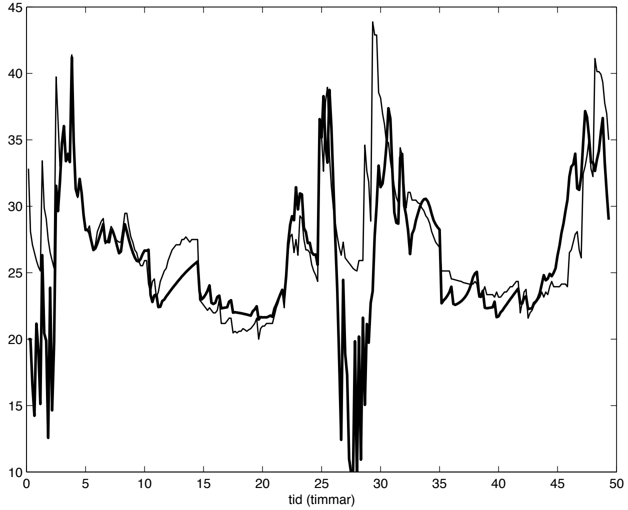
Eliminate $x(t)$:

$$\begin{aligned} y(t) =& (1+d_1)y(t-1) + (1-d_3)\frac{y(t-1)u(t-1)}{u(t-2)} \\ &+ (d_3-1)(1+d_1)\frac{y(t-2)u(t-1)}{u(t-2)} + d_0 d_2 u(t-1) \cdot I(t-2) \\ &- d_0 u(t-1)y(t-1) + d_0(1+d_1)u(t-1)y(t-2) \end{aligned}$$

Reparameterize with θ being the coefficiente above, ignoring links between them.



Neural Network Model



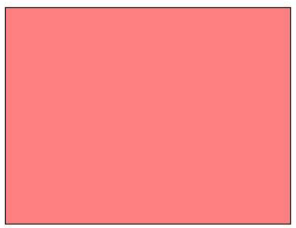
Nonlinear models - Outline

- General aspects
- Black-box models
- Grey-box Models
  - Physical Modeling
  - Semi-physical Modeling
  - Block-oriented models
  - Local Linear Models

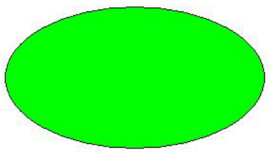
Block-oriented Models

→ 143

Building Blocks:

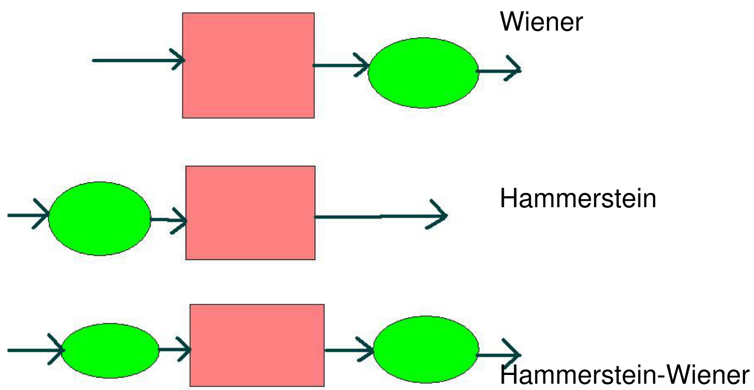


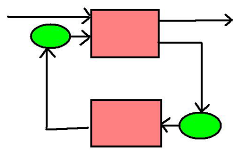
Linear Dynamic System $G(s)$



Nonlinear static function $f(u)$

Common Block Oriented Models

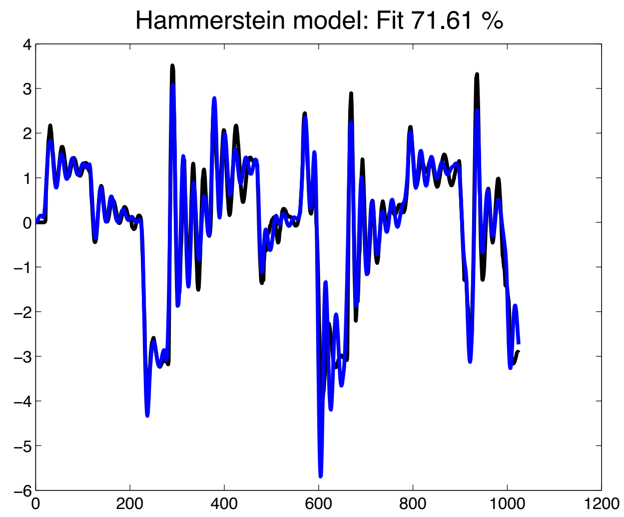




With the linear blocks parameterized as a linear dynamic system and the static blocks parameterized as a function ("curve"), this gives a parameterization of the output as

$$\hat{y}(t|\theta) = g(Z^{t-1}, \theta)$$

and the general approach of model fitting can be applied. However, in this contexts many algorithmic variants have been suggested.



- General aspects
- Black-box models
- Grey-box Models
  - Physical Modeling
  - Semi-physical Modeling
  - Block-oriented models
  - Local Linear Models

Non-linear systems are often handled by linearization around a working point:

$$\dot{x} = f(x,u); \qquad x^*, u^*, f(x^*, u^*) = 0$$
$$\Delta x = x - x^* \quad \Delta u = u - u^*$$
$$\dot{\Delta x} = f'_x(x^*, u^*)\Delta x + f'_u(x^*, u^*)\Delta u = A\Delta x + B\Delta u$$

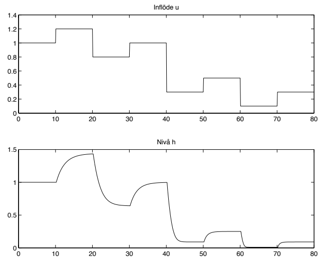
The idea behind Local Linear Models is to deal with the nonlinearities by selecting or averaging over relevant linearized models.

Example: Tank with Free Outflow

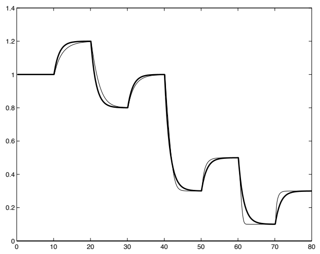
Example: Tank with inflow u and free outflow y and level h:
Equations (Bernoulli's law):

$$\dot{h} = -\sqrt{h} + u$$
$$y = \sqrt{h}$$

Experiment:



And linear model:



Tank Model: Linearization

Linearize around level h^* with corresponding flows $u^* = y^* = \sqrt{h^*}$:

$$\dot{h} = -\frac{1}{2\sqrt{h^*}}(h - h^*) + (u - u^*)$$
$$y = y^* + \frac{1}{2\sqrt{h^*}}(h - h^*)$$

First order linear system. In discrete time:

$$y(t) + \alpha_{h^*} y(t - T_s) = \beta_{h^*} u(t - T_s) + \gamma_{h^*} \qquad T_s = \text{sampling time}$$

Local Linear Tank Model

Sampled data model around level h^*:

$$\hat{y}_{h^*}(t) = \varphi^T(t)\theta_{h^*}$$
$$\varphi(t) = \begin{bmatrix} 1 & -y(t - T_s) & u(t - T_s) \end{bmatrix}^T$$
$$\theta_{h^*} = \begin{bmatrix} \gamma_{h^*} & \alpha_{h^*} & \beta_{h^*} \end{bmatrix}^T$$

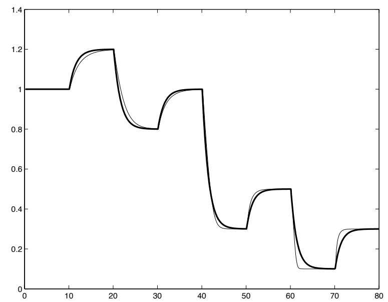
Total model: select or average over these local predictions, computed at a grid of values of h^* ($h_k : \ k = 1, \ldots, d$)

$$\hat{y}(t) = \sum_{k=1}^{d} w(h(t), h_k)\hat{y}_{h_k}(t) = \sum_{k=1}^{d} w(h(t), h_k)\varphi^T(t)\theta_{h_k}$$

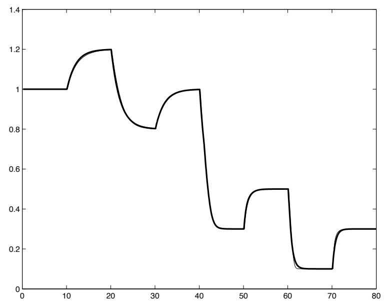
Choices of weighting function w : ....

Local Linear Models

Two models (d=2)



Five models ($d = 5$)



Let the measured working point variable (tank level in example) be denoted by $\rho(t)$ (sometimes called regime variable). If the regime variable is partitioned into d values ρ_k, the predicted output will be

$$\hat{y}(t) = \sum_{k=1}^{d} w(\rho(t), \rho_k)\hat{y}^{(k)}(t)$$

If the prediction $\hat{y}^{(k)}(t)$ corresponding to ρ_k is linear in the parameters, $\hat{y}^{(k)}(t) = \varphi^T(t)\theta^{(k)}$ the whole model will be a linear regression.

.

To build the model, we need to

- Select the regime variable ρ
- Decide the partition of the regime variable $w(\rho(t), \eta)$. Here $\eta = \{\rho_k; k = 1, \ldots, d\}$ is a parameter that describes the partition
- Find the local models in each partition.

If the local models are linear regressions, the total model will be

$$\hat{y}(t, \theta, \eta) = \sum_{k=1}^{d} w(\rho(t), \rho_k)\varphi^T(t)\theta^{(k)}$$

which for fixed partition η is a linear regression.

$$\hat{y}(t, \theta, \eta) = \sum_{k=1}^{d} w(\rho(t), \rho_k)\varphi^T(t)\theta^{(k)}$$

is also an example of a hybrid model (piecewise linear). If the partition is to be estimated too, the problem is considerably more difficult.

So called Linear Parameter Varying (LPV) are also closely related:

$$\dot{x} = A(\rho(t))x + B(\rho(t))u$$
$$y = C(\rho(t))x + D(\rho(t))u$$

Typical example: Aircraft, ρ being velocity and altitude.

- A nonlinear model can be seen as nonlinear mapping from past data to the space where the output lives: $\hat{y}(t|\theta) = g(Z^{t-1}, \theta, t)$. (Nonlinear i Z, but¸ could be linear i θ)
- Useful split of mapping: $g(Z^{t-1}, \theta) = g(\varphi(Z^{t-1}), \theta)$
- Black-box parameterizations, like ANN, usually employ one basic basis-function, that is scaled and located at different points
- Grey-boxes can be based on (serious) physical modeling and on more leisurely semi-physical modeling. Block-oriented or Local linear Models are other common model types.
- Non-convexity of the optimization remains one of the more serious problems for most parametric methods.